

# A Time and Space Efficient Algorithm for Contextual Linear Bandits

**José Bento**

Stanford University  
jbento@stanford.edu

**Stratis Ioannidis**

Technicolor  
stratis.ioannidis@technicolor.com

**S. Muthukrishnan**

Rutgers University  
muthu@cs.rutgers.edu

**Jinyun Yan**

Rutgers University  
jinyuny@cs.rutgers.edu

## Abstract

We consider a multi-armed bandits problem where payoffs are a linear function of an observed stochastic contextual variable. In the scenario where there exists a gap between optimal and suboptimal rewards, several algorithms have been proposed that achieve  $O(\log T)$  regret after  $T$  time steps. However, proposed methods either have a computation complexity per iteration that scales linearly with  $T$  or achieve regrets that grow linearly with the number of contexts. We propose an  $\epsilon$ -greedy type of algorithm that solves both limitations. In particular, when contexts are variables in  $\mathbb{R}^d$ , we prove that our algorithm has a constant computation complexity per iteration of  $O(\text{poly}(d))$  and can achieve a regret of  $O(\text{poly}(d) \log T)$  even when  $|\mathcal{X}| = \Omega(2^d)$ . In addition, unlike previous algorithms, its space complexity does not grow with  $T$ .

## 1 Introduction

The contextual multi-armed bandit problem is a sequential learning problem. At each time step, a learner has to choose among a set of possible actions/arms  $\mathcal{A}$ . Prior to making its decision, the learner observes some additional side information  $x \in \mathcal{X}$  over which he has no influence. This is commonly referred to as the *context*.

In general, the reward of a particular arm  $a \in \mathcal{A}$  under context  $x \in \mathcal{X}$  follows some unknown distribution. The goal of the learner is to select arms so that it minimizes its expected *regret*, *i.e.*, the expected difference between its cumulative reward and the reward accrued by an optimal policy, that knows the reward distributions.

Langford and Zhang (2007) propose an algorithm called *epoch-Greedy* for general contextual bandits. Their algorithm achieves an  $O(\log T)$  regret in the number of timesteps  $T$  in the *stochastic* setting, in which contexts are

sampled from an unknown distribution in an i.i.d. fashion. Unfortunately, the proposed algorithm and subsequent improvements (Dudik et al., 2011) have high computational complexity. Selecting an arm at time step  $t$  requires making a number of calls to a so-called *optimization oracle* that grows polynomially in  $T$ . In addition, implementing this optimization oracle can have a cost that grows linearly in  $|\mathcal{X}|$  in the worst case; this is prohibitive in many interesting cases, including the case where  $|\mathcal{X}|$  is exponential in the dimension of the context. In addition, both algorithms proposed in Langford and Zhang (2007) and Dudik et al. (2011) require keeping a history of observed contexts and arms chosen at every time instant. Hence, their space complexity grows linearly in  $T$ .

In this paper, we show that the challenges above can be addressed when rewards are linear. In the above contextual bandit set up, this means that  $\mathcal{X}$  is a subset of  $\mathbb{R}^d$ , and the expected reward of an arm  $a \in \mathcal{A}$  is an unknown linear function of the context  $x$ , *i.e.*, it has the form  $x^\top \theta_a$ , for some unknown vector  $\theta_a$ . This is a case of great interest, arising naturally when, conditioned on  $x$ , rewards from different arms are uncorrelated.

**Example 1. (Processor Scheduling)** A simple example is assigning incoming jobs to a set of processors  $\mathcal{A}$ , whose processing capabilities are not known *a priori*. This could be the case if, *e.g.*, the processors are machines in the cloud or, alternatively, humans offering their services through, *e.g.*, Mechanical Turk. Each arriving job is described by a set of attributes  $x \in \mathbb{R}^d$ , each capturing the work load of different types of sub-tasks this job entails, *e.g.*, computation, I/O, network communication, *etc.* Each processor's unknown feature vector  $\theta_a$  describes its processing capacity, *i.e.*, the time to complete a sub-task unit, in expectation. The expected time to complete a task  $x$  is given by  $x^\top \theta_a$ ; the goal of minimizing the delay (or, equivalently, maximizing its negation) brings us in the above bandit setting.  $\square$

**Example 2. (Group Activity Selection)** Another motivating example is maximizing group ratings, observed as the

outcome of a secret ballot election. In this setup, a subset of  $d$  users congregate to perform a joint activity, such as, *e.g.*, dining, rock climbing, watching a movie, *etc.* The group is dynamic and, at each timestep  $t \in \mathbb{N}$ , the vector  $x \in \{0, 1\}^d$ , is an indicator of present participants. An arm (*i.e.*, a joint activity) is selected; at the end of the activity, each user votes whether they liked the activity or not in a secret ballot, and the final tally is disclosed. In this scenario, the unknown vectors  $\theta_a \in \mathbb{R}^d$  indicate the probability a given participant will enjoy activity  $a$ , and the goal is to select activities that maximize the aggregate satisfaction among participants present at the given timestep.  $\square$

Our contributions are as follows.

- We isolate and focus on linear payoff case of stochastic multi-armed bandit problems, and design a simple arm selection policy which does not recourse to sophisticated oracles inherent in prior work.
- We prove that our policy achieves an  $O(\log T)$  regret after  $T$  steps in the stochastic setting, when the expected rewards of each arm are well separated. This meets the regret bound of best known algorithms for contextual multi-armed bandit problems.
- We show that our algorithm has  $O(|\mathcal{A}|d^2)$  computational complexity per step and its expected space complexity scales like  $O(|\mathcal{A}|d^2)$ . This is a significant improvement over known contextual multi-armed bandit problems, as well as for bandits specialized for linear payoffs.

Our algorithm is inspired by the work of Auer et al. (2002) on the  $\epsilon$ -greedy algorithm and the use of linear regression to estimate the parameters  $\theta_a$ . The main technical innovation is the use of matrix concentration bounds to control the error of the estimates of  $\theta_a$  in the stochastic setting. We believe that this is a powerful realization and may ultimately help us analyze richer classes of payoff functions.

The remainder of this paper is organized as follows: in Section 2 we compare our results with existing literature. In Section 3 we describe the set up of our problem in more detail. In the Section 4 we state our main results and in Section 5 we discuss them. Section 6 is devoted to exemplifying the performance and limitations of our algorithm by means of simple numerical simulations. In Section 7 we prove our results. Finally, in Section 8 we draw our conclusions.

## 2 Related Work

The original paper by Langford and Zhang (2007) assumes that, conditioned on the arm and the context, rewards are sampled from a probability distribution  $p_{a,x}$ . As is common in bandit problems, there is a tradeoff between *explo-*

*ration*, *i.e.*, selecting arms  $a \in \mathcal{A}$  to sample rewards from the distributions  $p_{a,x}$  and learn about them, and *exploitation*, whereby knowledge of these distributions based on the samples is used to select an arm that yields a high payoff.

A significant challenge is that during the exploitation phase, conditioned on the fact that an arm  $a$  was chosen, the distribution of observed contexts does not follow  $p(x|a)$ . In fact, an arm will tend to be selected more often in contexts in which it performs well. The epoch-Greedy algorithm deals with this by separating the exploration and exploitation phase, effectively selecting an arm uniformly at random at certain time slots (the exploration “epochs”), and using samples collected only during these epochs to estimate the payoff of each arm in the remaining time slots (for exploitation). Langford and Zhang (2007) establish an  $O(T^{2/3}(\ln |\mathcal{X}|)^{1/3})$  on the regret for epoch-Greedy in the stochastic setting. They further improve this to  $O(\log T)$  when a lower bound on the gap between optimal and sub-optimal arms in each context exists.

Unfortunately, the price of the generality of the framework of Langford and Zhang (2007) is the high computational complexity when selecting an arm during an exploitation phase. In a recent improvement (Dudik et al., 2011), this computation requires a *poly*( $t$ ) number of calls to an optimization oracle. Most importantly, even in the linear case we study here, there is no clear way to implement this oracle in sub-exponential time in  $d$ , the dimension of the context. As Dudik et al. (2011) point out, the optimization oracle solves a so-called cost-sensitive classification problem. In the particular case of linear bandits, the oracle thus reduces to finding the “least-costly” linear classifier. This is hard, even in the case of only two arms: finding the linear classifier with the minimal number of errors is NP-hard (Johnson and Preparata, 1978), and remains NP hard even if an approximate solution is required (Bartlett and Ben-David, 1999).

As such, a different approach is warranted under linear rewards. On the other hand, linear bandits have been extensively studied in the following setup, which is more general than ours. In the classic linear bandit setup, the arms themselves are represented as vectors, *i.e.*,  $\mathcal{A} \subset \mathbb{R}^d$ , and, in addition, the set  $\mathcal{A}$  can change from one time slot to the next. The expected payoff of an arm  $a$  with vector  $x_a$  is given by  $x_a^\top \theta$ , for some unknown vector  $\theta \in \mathbb{R}^d$ , common among all arms.

There are several different variants of the above model. Auer (2002) and, more recently, Li et al. (2010) and Chu et al. (2011), study this problem in the adversarial setting. In particular,  $|\mathcal{A}|$  is fixed (and finite) and  $\mathcal{A} \subset \mathbb{R}^d$  is given at each time by an adversary that has full knowledge of what the learner knows, but cannot a priori predict the outcome

of any random variables before the learner observes them. Dani et al. (2008), Rusmevichientong and Tsitsiklis (2010), and Abbasi-Yadkori et al. (2011) study the problem in the stochastic setting, in cases where  $\mathcal{A}$  is a fixed but possibly uncountable bounded subset of  $\mathbb{R}^d$ .

The regret bounds on all of the above setups (both stochastic and adversarial) are of the order of  $O(\sqrt{T} \text{polylog}(T))$ . An important distinction between the aforementioned general linear bandit setup and the contextual model of Langford and Zhang (2007) (and, *a fortiori*, our model as well) is that in the above setting, different arms' payoffs are *correlated*. Payoffs observed for any arm inform the learner about the common unknown  $\theta$  and, hence, help infer the payoff of a different arm. Exploiting this correlation to achieve low regret constitutes the main challenge of the above setups. In Langford and Zhang (2007), as in our case, the reward of an arm does not reveal any information about the reward of another arm. Nevertheless, the rewards for the *same* arm under different contexts are correlated. Exploiting this correlation to learn the unknown vectors  $\theta_a$  faster and achieve low regret constitutes the main challenge of our setup.

Note that our problem can be expressed as a special case of the above linear bandits setup by taking  $\theta = [\theta_1; \dots; \theta_K] \in \mathbb{R}^{Kd}$ , where  $K = |\mathcal{A}|$ , and, given context  $x$ , associating the  $i$ -th arm with an appropriate vector of the form  $x_{a_i} = [0 \dots x \dots 0]$ . As such, all of the above  $O(\sqrt{T} \text{polylog}(T))$  bounds (and respective algorithms) can be applied to our setup. However, these algorithms do not exploit the fact that, in our setting, arms are uncorrelated. We exploit this to obtain a logarithmic regret, for a much simpler algorithm than the ones outlined in the above works.

### 3 Model

In this section, we give a precise definition of the linear contextual bandit problem we study in this work.

#### 3.1 Contexts

At every time instant  $t \in \{1, 2, \dots\}$ , a context  $x_t \in \mathcal{X} \subset \mathbb{R}^d$ , is observed by the learner. We assume that  $\|x\|_2 \leq 1$ ; as the expected reward is linear in  $x$ , this assumption is w.l.o.g. We prove our main result (Theorem 2) in the stochastic setting where  $x_t$  are drawn i.i.d. from an unknown multivariate probability distribution  $\mathcal{D}$ .

In addition, we require that the set of contexts is finite *i.e.*,  $|\mathcal{X}| \leq \infty$ . We define  $\Sigma_{\min} > 0$  to be the smallest non-zero eigenvalue of the covariance matrix  $\Sigma \equiv \mathbb{E}\{x_1 x_1^\dagger\}$ .

#### 3.2 Arms and actions

At time  $t$ , after observing the context  $x_t$ , the learner decides to play an arm  $a \in \mathcal{A}$ , where  $K \equiv |\mathcal{A}|$  is fi-

nite. We denote the arm played at this time by  $a_t$ . We study *adaptive* arm selection policies, whereby the selection of  $a_t$  depends only on the current context  $x_t$ , and on all past contexts, actions and rewards. In other words,  $a_t = a_t(x_t, \{x_\tau, a_\tau, r_\tau\}_{\tau=1}^{t-1})$ .

#### 3.3 Payoff

After observing a context  $x_t$  and selecting an arm  $a_t$ , the learner receives a payoff  $r_{a_t, x_t}$  which is drawn from a distribution  $p_{a_t, x_t}$  independently of all past contexts, actions or payoffs. We assume that the expected payoff is a linear function of the context. In other words,

$$r_{a_t, x_t} = x_t^\dagger \theta_a + \epsilon_{a, t} \quad (1)$$

where  $\{\epsilon_{a, t}\}_{a \in \mathcal{A}, t \geq 1}$  are a set of independent random variables with zero mean and  $\{\theta_a\}_{a \in \mathcal{A}}$  are unknown parameters in  $\mathbb{R}^d$ . Note that, w.l.o.g, we can assume that  $Q = \max_{a \in \mathcal{A}} \|\theta_a\|_2 \leq 1$ . This is because if  $Q > 1$ , as payoffs are linear, we can divide all payoffs by  $Q$ ; the resulting payoff is still a linear model, and our results stated below apply.

Recall that  $Z$  is a sub-gaussian random variable with constant  $L$  if  $\mathbb{E}\{e^{\gamma Z}\} \leq e^{\gamma^2 L^2}$ . In particular, sub-gaussianity implies  $\mathbb{E}\{Z\} = 0$ . We make the following technical assumption.

**Assumption 1** *The random variables  $\{\epsilon_{a, t}\}_{a \in \mathcal{A}, t \geq 1}$  are sub-gaussian random variables with constant  $L > 0$ .*

#### 3.4 Regret

Given a context  $x$ , the arm that gives highest expected reward is

$$a_x^* = \arg \max_{a \in \mathcal{A}} x^\dagger \theta_a. \quad (2)$$

The expected cumulative regret the learner experiences over  $T$  steps is defined by,

$$R(T) = \mathbb{E} \left\{ \sum_{t=1}^T x_t^\dagger (\theta_{a_{x_t}^*} - \theta_{a_t}) \right\}. \quad (3)$$

The expectation above is taken over the contexts  $x_t$ .

The objective of the learner is to design a policy  $a_t = a_t(x_t, \{x_\tau, a_\tau, r_\tau\}_{\tau=1}^{t-1})$  that achieves as low expected cumulative regret as possible. In this paper we are also interested in arm selection policies having a low computational complexity.

We define  $\Delta_{\max} \equiv \max_{a, b \in \mathcal{A}} \|\theta_a - \theta_b\|_2$ , and

$$\Delta_{\min} \equiv \inf_{x \in \mathcal{X}, a: x^\dagger \theta_a < x^\dagger \theta_{a_x^*}} x^\dagger (\theta_{a_x^*} - \theta_a) > 0$$

Observe that, by the finiteness of  $\mathcal{X}$  and  $\mathcal{A}$ , the above infimum is attained (*i.e.*, it is a minimum) and is indeed positive.

## 4 Main results

We now present a simple and *efficient* on-line algorithm that, under the above assumptions, has expected *logarithmic* regret. Specifically, its computational complexity, at each time instant, is  $O(Kd^2)$  and the expected memory requirement scales like  $O(Kd^2)$ . As far as we know, our analysis is the first to show that a simple and *efficient* algorithm for the problem of linearly parametrized bandits can, under reward separation and i.i.d. contexts, achieve logarithmic expected cumulative regret.

Before we present our algorithm in full detail, let us give some intuition about it. Part of the job of the learner is to estimate the unknown parameters  $\theta_a$  based on past actions, contexts and rewards. We denote the estimate of  $\theta_a$  at time  $t$  by  $\hat{\theta}_{a,t}$ . If  $\theta_a \approx \hat{\theta}_{a,t}$  then, given an observed context, the learner will more accurately know which arm to play to incur in small regret. The estimates  $\hat{\theta}_{a,t}$  can be constructed based on a history of past rewards, contexts and arms played.

Since observing a reward  $r$  for arm  $a$  under context  $x$  does not give information about the magnitude of  $\theta_a$  along directions orthogonal to  $x$ , it is important that, for each arm, rewards are observed and recorded for a rich class of contexts. This gives rise to the following challenge: If the learner tries to build this history while trying to minimize the regret, the distribution of contexts observed when playing a certain arm  $a$  will be biased and potentially not rich enough. In particular, when trying to achieve a small regret, conditioned on  $a_t = a$ , it is more likely that  $x_t$  is a context for which  $a$  is optimal.

We address this challenge using the following idea, also appearing in the epoch-Greedy algorithm of Langford and Zhang (2007). We partition time slots into *exploration* and *exploitation epochs*. In exploration epochs, the learner plays arms uniformly at random, independently of the context, and records the observed rewards. This guarantees that in the history of past events, each arm has been played along with a sufficiently rich set of contexts. In exploitation epochs, the learner makes use of the history of events stored during exploration to estimate the parameters  $\theta_a$  and determine which arm to play given a current observed context. The rewards observed during exploitation are not recorded.

More specifically, when exploiting, we learner performs two operations. In the first operation, for each arm  $a \in \mathcal{A}$ , an estimate  $\hat{\theta}_a$  of  $\theta_a$  is constructed from a simple  $\ell_2$ -regularized regression, as in in Auer (2002) and Li et al. (2010). In the second operation, the learner plays the arm  $a$  that maximizes  $x_t^\dagger \hat{\theta}_a$ . Crucially, in the first operation, only information collected during exploration epochs is used. In particular, let  $\mathcal{T}_{a,t-1}$  be the set of exploration epochs up to and including time  $t - 1$  (i.e., the times that the learner

played an arm u.a.r.). Moreover, for any  $\mathcal{T} \in \mathbb{N}$ , denote by  $r_{\mathcal{T}} \in \mathbb{R}^n$  is a vector of observed rewards for all time instances  $t \in \mathcal{T}$ , and  $X_{\mathcal{T}} \in \mathbb{R}^{n \times d}$  is a matrix of  $\mathcal{T}$  rows, each containing one of the observed contexts at time  $t \in \mathcal{T}$ . Then, at timeslot  $t$  the estimator  $\hat{\theta}_a$  is the solution of the following convex optimization problem.

$$\min_{\theta \in \mathbb{R}^d} \frac{1}{2n} \|r_{\mathcal{T}} - X_{\mathcal{T}}\theta\|_2^2 + \frac{\lambda_n}{2} \|\theta\|_2^2. \quad (4)$$

where  $\mathcal{T} = \mathcal{T}_{a,t-1}$ ,  $n = |\mathcal{T}_{a,t-1}|$ ,  $\lambda_n = 1/\sqrt{n}$ . In other words, the estimator  $\hat{\theta}_a$  is a (regularized) estimate of  $\theta_a$ , based only on observations made during exploration epochs. Note that the solution to (4) is given by

$$\hat{\theta}_a = \left( \lambda_n I + \frac{1}{n} X_{\mathcal{T}}^\dagger X_{\mathcal{T}} \right)^{-1} \frac{1}{n} X_{\mathcal{T}}^\dagger r_{\mathcal{T}} \quad (5)$$

---

### Algorithm 1 Contextual $\epsilon$ -greedy

---

For all  $a \in \mathcal{A}$ , set  $A_a \leftarrow 0_{d \times d}$ ;  $n_a \leftarrow 0$ ;  $b_a \leftarrow 0_d$

**for**  $t = 1$  to  $p$  **do**

$a \leftarrow 1 + (t \bmod K)$

Play arm  $a$

$n_a \leftarrow n_a + 1$ ;  $b_a \leftarrow b_a + r_t x_t$ ;  $A_a \leftarrow A_a + x_t x_t^\dagger$

**end for**

**for**  $t = p + 1$  to  $T$  **do**

$e \leftarrow \text{Bernoulli}(p/t)$

**if**  $e = 1$  **then**

$a \leftarrow \text{Uniform}(1/K)$

Play arm  $a$

$n_a \leftarrow n_a + 1$ ;  $b_a \leftarrow b_a + r_t x_t$ ;  $A_a \leftarrow A_a + x_t x_t^\dagger$

**else**

**for**  $a \in \mathcal{A}$  **do**

Get  $\hat{\theta}_a$  as the solution to the linear system:

$$\left( \lambda_{n_a} I + \frac{1}{n_a} A_a \right) \hat{\theta}_a = \frac{1}{n_a} b_a$$

**end for**

Play arm  $a = \arg \max_b x_t^\dagger \hat{\theta}_b$

**end if**

**end for**

---

An important design choice is the above process selection of the time slots at which the algorithm explores, rather than exploits. Following the ideas of Sutton and Barto (1998), we select the exploration epochs so that they occur approximately  $\Theta(\log t)$  times after  $t$  slots. This guarantees that, at each time step, there is enough information in our history of past events to determine the parameters accurately while only incurring in a regret of  $O(\log t)$ . There are several ways of achieving this; our algorithm explores at each time step with probability  $\Theta(t^{-1})$ .

The above steps are summarized in pseudocode by Algorithm 1. Note that the algorithm contains a scaling parameter  $p$ , which is specified below, in Theorem 2. Because there are  $K$  arms and for each arm  $(x_t, r_{a,t}) \in \mathbb{R}^{d+1}$ ,



the expected memory required by the algorithm scales like  $O(Kd^2)$ . In addition, both the matrix  $X_{\mathcal{T}}^\dagger X_{\mathcal{T}}$  and the vector  $X_{\mathcal{T}}^\dagger r_{\mathcal{T}}$  can be computed in an online fashion in  $O(d^2)$  time:  $X_{\mathcal{T}}^\dagger X_{\mathcal{T}} \leftarrow X_{\mathcal{T}}^\dagger X_{\mathcal{T}} + x_t x_t^\dagger$  and  $X_{\mathcal{T}}^\dagger r_{\mathcal{T}} \leftarrow X_{\mathcal{T}}^\dagger r_{\mathcal{T}} + r_t x_t$ . Finally, the estimate of  $\hat{\theta}_a$  does not require full matrix inversion but only solving a linear system (see Algorithm 1), which can be done in  $O(d^2)$  time. The above is summarized in the following theorem,

**Theorem 1** *Algorithm 1 has computational complexity of  $O(Kd^2)$  per iteration and its expected space complexity scales like  $O(Kd^2)$ .*

We now state our main theorem that shows that Algorithm 1 achieves  $R(T) = O(\log T)$ .

**Theorem 2** *Under Assumptions 1, the expected cumulative regret of algorithm 1 satisfies,*

$$R(T) \leq p\Delta_{\max}\sqrt{d} + 14\Delta_{\max}\sqrt{d}Ke^{Q/4} + p\Delta_{\max}\sqrt{d}\log T.$$

for any

$$p \geq \frac{CKL'^2}{(\Delta'_{\min})^2(\Sigma'_{\min})^2}. \quad (6)$$

Above,  $C$  is a universal constant,  $\Delta'_{\min} = \min\{1, \Delta_{\min}\}$ ,  $\Sigma'_{\min} = \min\{1, \Sigma_{\min}\}$  and  $L' = \max\{1, L\}$ .

Algorithm 1 requires the specification of the constant  $p$ . In Section 4.2, we give two examples of how to efficiently choose a  $p$  that satisfies (6).

In Theorem 2, the bound on the regret depends on  $p$  - small  $p$  is preferred - and hence it is important to understand how the r.h.s. of (6) might scale when  $K$  and  $d$  grow. In Section 4.1, we show that, for a concrete distribution of contexts and choice of expected rewards  $\theta_a$ , and assuming (6) holds,  $p = O(K^3 d^5)^{-1}$ . There is nothing special about the concrete details of how contexts and  $\theta_a$ 's are chosen and, although not included in this paper, for many other distributions, one also obtains  $p = O(\text{poly}(d))$ . We can certainly construct pathological cases where, for example,  $p$  grows exponentially with  $d$ . However, we do not find these intuitive. Specially when interpreting these having in mind real applications as the ones introduced in Example 1 and Example 2.

#### 4.1 Example of scaling of $p$ with $d$ and $K$

Assume that contexts are obtained by normalizing a  $d$ -dimensional vector with i.i.d. entries as Bernoulli random variables with parameter  $w$ . Assume in addition that every  $\theta_a$  is obtained i.i.d. from the following prior distribution:

<sup>1</sup>This bound holds with probability converging to 1 as  $K$  and  $d$  get large

every entry of  $\theta_a$  is drawn i.i.d. from a uniform distribution and then  $\theta_a$  is normalized. Finally, assume that the payoffs are given by  $r_{a,t} = x_t^\dagger \Theta_a$ , where  $\Theta_a \in \mathbb{R}^d$  are random variables that fluctuate around  $\theta_a = \mathbb{E}\{\Theta_a\}$  with each entry fluctuating by at most  $F$ .

Under these assumptions the following is true:

- $\Sigma_{\min} = \Omega(d^{-1})$ . In fact, the same result holds asymptotically independently of  $w = w(d)$  if, for example, we assume that on average groups are roughly of the same size,  $M$ , with  $w = M/d$ ;
- $L = O(\sqrt{d})$ . This holds because  $\epsilon_{a,t} = r_{a,t} - \mathbb{E}\{r_{a,t}\} = x_t^\dagger(\Theta_a - \theta_a)$  are bounded random variables with zero mean and  $\|x_t^\dagger(\Theta_a - \theta_a)\|_\infty = O(\sqrt{d})$ .
- $\Delta_{\min} = \Omega(1/(Kd\sqrt{w}))$  with high-probability (for large  $K$  and  $d$ ). This can be seen as follows, if  $\Delta_{\min} = x^\dagger(\theta_a - \theta_b)$  for some  $x, a$  and  $b$ , then it must be true that  $\theta_a$  and  $\theta_b$  differ in a component for which  $x$  is non-zero. The minimum difference between components among all pairs of  $\theta_a$  and  $\theta_b$  is lower bounded by  $\Omega(1/(K\sqrt{d}))$  with high probability (for large  $K$  and  $d$ ). Taking into account that each entry of  $x$  is of order  $O(1/\sqrt{dw})$  with high-probability, the bound on  $\Delta_{\min}$  follows.

If we want to apply Theorem 2 then (6) must hold and hence putting all the above calculations together we conclude that  $p = O(K^3 d^5)$ .

#### 4.2 Computing $p$ in practice

If we have knowledge of an a priori distribution for the contexts, for the expected payoffs and for the variance of the rewards then we can quickly compute the value of  $\Sigma_{\min}$ ,  $L$  and a typical value for  $\Delta_{\min}$ . An example of this was done above (Section 4.1). There, the values were presented only in order notation but exact values are not hard to obtain for that and other distributions. Since a suitable  $p$  only needs to be larger than the r.h.s. of (6), by introducing an appropriate multiplicative constant, we can produce a  $p$  that satisfied (6) with high probability.

If we have no knowledge of any model for the contexts or expected payoffs, it is still possible to find  $p$  by estimating  $\Delta_{\min}$ ,  $\Sigma_{\min}$  and  $L$  from data gathered while running Algorithm 1. Notice again that, since all that is required for our theorem to hold is that  $p$  is greater than a certain function of these quantities, an exact estimation is not necessary. This is important because, for example, accurately estimating  $\Sigma_{\min}$  is hard when matrix  $\mathbb{E}\{x_1 x_1^\dagger\}$  has a large condition number.

Not being too concerned about accuracy,  $\Sigma_{\min}$  can be estimated from  $\mathbb{E}\{x_1 x_1^\dagger\}$ , which can be estimated from the

sequence of observed  $x_t$ .  $\Delta_{\min}$  can be estimated from Algorithm 1 by keeping track of the smallest difference observed until time  $t$  between  $\max_b x_t^\top \hat{\theta}_b$  and the second largest value of the function being maximized. Finally, the constant  $L$  can be estimated from the variance of the observed rewards for the same (or similar) contexts. Together, these estimations do not incur in any significant loss in computational performance of our algorithm.

## 5 Adversarial setting

In the stochastic setting, the richness of the subset of  $\mathbb{R}^d$  spanned by the observed contexts is related to the skewness of the distribution  $\mathcal{D}$ . The fact that our result depends on  $\Sigma_{\min}$  and that the regret increases as this value becomes smaller indicates that we are still far from proving the  $O(\log T)$  regret for the adversarial setting, where an adversary chooses the contexts.

In particular, the main difficulty in using a linear regression, and the reason why our result depends on  $\Sigma_{\min}$ , is related to the dependency of our estimation of  $x_t^\top \theta_a$  on  $\frac{1}{|\mathcal{T}_{a,t-1}|} X_{\mathcal{T}_{a,t-1}}^\top X_{\mathcal{T}_{a,t-1}}$ . It is not hard to show that the error in approximating  $x_t^\top \theta_a$  with  $x_t^\top \hat{\theta}_a$  is proportional to

$$\sqrt{x_t^\top \left( \lambda_n I + \frac{1}{n} X_{\mathcal{T}}^\top X_{\mathcal{T}} \right)^{-2}} x_t. \quad (7)$$

Looking at this expression one quickly sees that, even if a given context has been observed relatively often in the past, the algorithm can “forget” it because of the *mean* over contexts that is being used to produce our estimates of  $x_t^\top \theta_a$ .

The effect of this phenomenon on the performance of Algorithm 1 can be readily seen in the following pathological example. Assume that  $\mathcal{X} = \{(1, 1), (1, 0)\} \subset \mathbb{R}^2$ . Assume that the contexts arrive in the following way:  $(1, 1)$  appears with probability  $1/I$  and  $(1, 0)$  appears with probability  $1 - 1/I$ . The correlation matrix for this stochastic process is  $\{(1, 1/I), (1/I, 1/I)\}$  and its minimum eigenvalue scales like  $O(1/I)$ . Hence, the regret scales as  $O(I^2 \log T)$ . If  $I$  is allowed to slowly grow with  $t$ , we expect that our algorithm will not be able to guarantee a logarithmic regret (assuming that our upper bound is tight). In other words, although  $(1, 1)$  might have appeared a sufficient number of times for us to be able to predict the expected reward for this context, Algorithm 1 performs poorly since the mean (7) will be ‘saturated’ with the context  $(1, 0)$  and forget about  $(1, 1)$ .

The solution for this problem is to ignore some past contexts when building an estimate for  $x_t^\top \theta_a$ , by including in the mean (7) past contexts that are closer in direction to the current context  $x_t$ . Having this in mind, and building on the ideas of Auer (2002), we propose the UCB-type Algorithm 2.

---

### Algorithm 2 Contextual UCB

---

```

for  $t = 1$  to  $p$  do
   $a \leftarrow 1 + (t \bmod K)$ 
  Play arm  $a$ 
   $\mathcal{T}_{a,t} \leftarrow \mathcal{T}_{a,t-1} \cup \{t\}$ 
end for
for  $t = p + 1$  to  $T$  do
  for  $a \in \mathcal{A}$  do
     $c_{a,t} \leftarrow \min_{\mathcal{T} \subset \mathcal{T}_{a,t-1}} \frac{\log t}{|\mathcal{T}|} x_t^\top \left( \lambda_n I + \frac{1}{n} X_{\mathcal{T}}^\top X_{\mathcal{T}} \right)^{-2} x_t$ 
     $\mathcal{T}^* \leftarrow$  subset of  $\mathcal{T}_{a,t-1}$  that achieves the minimum
     $n \leftarrow |\mathcal{T}^*|$ 
    Get  $\hat{\theta}_a$  as the solution to the linear system:
     $\left( \lambda_n I + \frac{1}{n} X_{\mathcal{T}^*}^\top X_{\mathcal{T}^*} \right) \hat{\theta}_a = \left( \frac{1}{n} X_{\mathcal{T}^*}^\top r_{\mathcal{T}^*} \right)$ 
  end for
  Play arm  $a = \arg \max_b x_t^\top \hat{\theta}_b + \sqrt{c_{b,t}}$ 
   $\mathcal{T}_{a,t} \leftarrow \mathcal{T}_{a,t-1} \cup \{t\}$ 
end for

```

---

It is straightforward to notice that this algorithm cannot be implemented in an efficient way. In particular, the search for  $\mathcal{T}^* \subset \mathcal{T}_{a,t-1}$  has a computational complexity exponential in  $t$ . The challenge is to find an efficient way of approximating  $\mathcal{T}^*$  efficiently. This can be done by either reducing the size of  $\mathcal{T}_{a,t-1}$  – the history from which one wants to extract  $\mathcal{T}_{a,t-1}$  – by not storing all events in memory<sup>2</sup> or by finding an efficient algorithm of approximating the minimization over the  $\mathcal{T}_{a,t-1}$  (or both). It remains an open problem to find such an approximation scheme and to prove that it achieves  $O(\log T)$  regret.

## 6 Numerical results

In Theorem 2, we showed that, in the stochastic setting, Algorithm 1 has an expected regret of order  $O(\log T)$ . We now illustrate this point by numerical simulations and, most importantly, exemplify how violating the stochastic assumption might degrade its performance.

Figure 1 shows the average cumulative regret (in semi-log scale) over 10 independent runs of Algorithm 1 for  $T = 10^5$  and for the following setup. The context variables  $x \in \mathbb{R}^3$  and at each time step  $\{x_t\}_{t \geq 1}$  are drawn i.i.d. in the following way: (a) set each entry of  $x$  to 1 or 0 independently with probability 1/2; (b) normalize  $x$ . We consider  $K = 6$  arms with corresponding parameters  $\theta_a$  generated independently from a standard multivariate gaussian distribution. Given a context  $x$  and an arm  $a$ , rewards were random and independently generated from a uniform distribution  $U([0, 2x^\top \theta_a])$ . As expected, the regret is logarithmic. Figure 1 shows a straight line at the end.

---

<sup>2</sup>For example, if we can guarantee that  $|\mathcal{T}_{a,t}| = O(\log t)$  then the complexity of the above algorithm at time step  $t$  is  $O(t)$ .

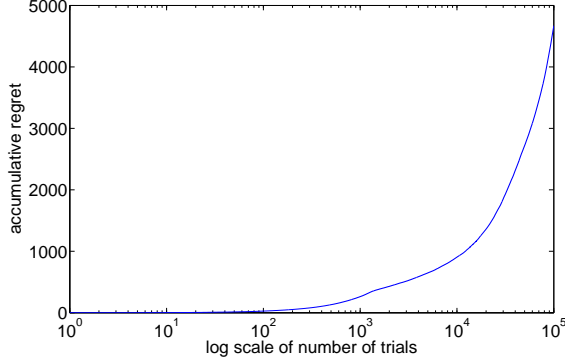


Figure 1: Regret over  $T$  when  $x_t$  is from i.i.d.

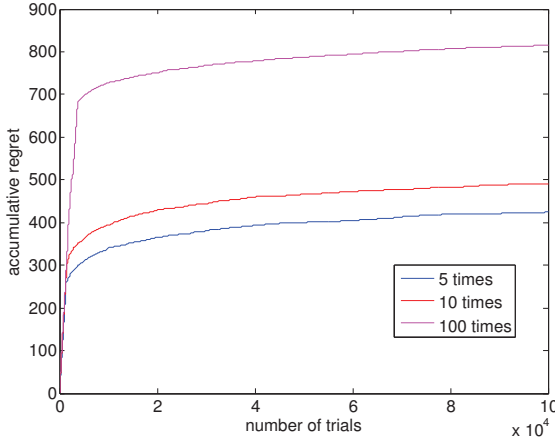


Figure 2: Regret over  $T$  when  $x_t$  is not from i.i.d.

To understand the effect of the stochasticity of  $x$  on the regret, we consider the following scenario: with every other parameter unchanged, let  $\mathcal{X} = \{x, x'\}$ . At every time step  $x = [1, 1, 1]$  appears with probability  $1/I$ , and  $x' = [1, 0, 1]$  appears with probability  $1 - (1/I)$ . Figure 2 shows the dependency of the expected regret on the context distribution for  $I = 5, 10$  and  $100$ . One can see that an increase of  $I$  causes a proportional increase in the regret.

## 7 Proofs

### 7.1 Proof of Theorem 2

The general structure of the proof of our main result follows that of Auer et al. (2002). The main technical innovation is the realization that, in the setting when the contexts are drawn i.i.d. from some distribution, a standard matrix concentration bound allows us to treat  $\lambda_n I + n^{-1}(X_{\mathcal{T}}^\top X_{\mathcal{T}})$  in Algorithm 1 as a deterministic positive-definite symmetric matrix, even as  $\lambda_n \rightarrow 0$ .

Let  $\mathcal{E}_T$  denote the time instances for  $t > p$  and until time  $T$

in which the algorithm took an exploitation decision. Recall that, by Cauchy-Schwarz inequality,  $x_t^\top(\theta_{a_{x_t}}^* - \theta_{a_t}) \leq \|x_t\|_1 \|(\theta_{a_{x_t}}^* - \theta_{a_t})\|_\infty \leq \sqrt{d} \|x_t\|_2 \|(\theta_{a_{x_t}}^* - \theta_{a_t})\|_\infty \leq \sqrt{d} \Delta_{\max}$ . In addition, recall that  $\sum_{t=2}^T 1/t \leq \log T$ . For  $R(T)$  the cumulative regret until time  $T$ , we can write,

$$\begin{aligned} R(T) &= \mathbb{E} \left\{ \sum_{t=1}^T x_t^\top (\theta_{a_{x_t}}^* - \theta_{a_t}) \right\} \leq p \Delta_{\max} \sqrt{d} \\ &\quad + \Delta_{\max} \sqrt{d} \mathbb{E} \left\{ \sum_{t=p+1}^T \mathbb{I}_{\{x_t^\top \theta_{a_t} < x_t^\top \theta_{a_{x_t}}^*\}} \right\} \\ &\leq p \Delta_{\max} \sqrt{d} + \Delta_{\max} \sqrt{d} \mathbb{E} \{ |\mathcal{E}_T| \} \\ &\quad + \Delta_{\max} \sqrt{d} \mathbb{E} \left\{ \sum_{t \in \mathcal{E}_T} \mathbb{I}_{\{x_t^\top \theta_{a_t} < x_t^\top \theta_{a_{x_t}}^*\}} \right\} \\ &\leq p \Delta_{\max} \sqrt{d} + p \Delta_{\max} \sqrt{d} \log T \\ &\quad + \Delta_{\max} \sqrt{d} \mathbb{E} \left\{ \sum_{t \in \mathcal{E}_T} \mathbb{I}_{\{x_t^\top \theta_{a_t} < x_t^\top \theta_{a_{x_t}}^*\}} \right\} \\ &\leq p \Delta_{\max} \sqrt{d} + p \Delta_{\max} \sqrt{d} \log T \\ &\quad + \Delta_{\max} \sqrt{d} \mathbb{E} \left\{ \sum_{t \in \mathcal{E}_T} \sum_{a \in \mathcal{A}} \mathbb{I}_{\{x_t^\top \hat{\theta}_{a,t} > x_t^\top \hat{\theta}_{a_{x_t}}^*, t\}} \right\}. \end{aligned}$$

In the last line we used the fact that when exploiting, if we do not exploit the optimal arm  $a_{x_t}^*$ , then it must be the case that the estimated reward for some arm  $a$ ,  $x_t^\top \hat{\theta}_{a,t}$ , must exceed that of the optimal arm,  $x_t^\top \hat{\theta}_{a_{x_t}}^*, t$ , for the current context  $x_t$ . Above, we use  $\hat{\theta}_{a,t}$  instead of  $\hat{\theta}_a$  to explicitly state the dependency of estimators on the stored history for each arm until time  $t$ .

We can continue the chain of inequalities and write,

$$\begin{aligned} R(T) &\leq p \Delta_{\max} \sqrt{d} + p \Delta_{\max} \sqrt{d} \log T \\ &\quad + \Delta_{\max} \sqrt{d} K \sum_{t=1}^T \mathbb{P} \{ x_t^\top \hat{\theta}_{a,t} > x_t^\top \hat{\theta}_{a_{x_t}}^*, t \}. \end{aligned}$$

The above expression depends on the value of the estimators for time instances that might or might not be exploitation times. For each arm, these are computed just like in Algorithm 1, using the most recent history available. The above probability depends on the randomness of  $x_t$  and on the randomness of recorded history for each arm.

Since  $x_t^\top(\theta_{a_{x_t}}^* - \theta_a) \geq \Delta_{\min}$  we can write

$$\begin{aligned} \mathbb{P} \{ x_t^\top \hat{\theta}_{a,t} > x_t^\top \hat{\theta}_{a_{x_t}}^*, t \} &\leq \mathbb{P} \left\{ x_t^\top \hat{\theta}_{a,t} \geq x_t^\top \theta_a + \frac{\Delta_{\min}}{2} \right\} \\ &\quad + \mathbb{P} \left\{ x_t^\top \hat{\theta}_{a_{x_t}}^*, t \leq x_t^\top \theta_{a_{x_t}}^* - \frac{\Delta_{\min}}{2} \right\}. \end{aligned}$$

We now bound each of these probabilities separately. Since their bound is the same, we focus only on the first probability.

Substituting the definition of  $r_{a,t} = x_t^\dagger \theta_a + \epsilon_{a,t}$  into the expression for  $\hat{\theta}_{a,t}$  one readily obtains,

$$(\hat{\theta}_{a,t} - \theta_a) = \left( \lambda_n I + \frac{1}{n} X_{\mathcal{T}}^\dagger X_{\mathcal{T}} \right)^{-1} \left( \frac{1}{n} \sum_{\tau \in \mathcal{T}} x_\tau \epsilon_{a,\tau} - \lambda_n \theta_a \right).$$

We are using again the notation  $\mathcal{T} = \mathcal{T}_{a,t-1}$  and  $n = |\mathcal{T}|$ . From this expression, an application of Cauchy-Schwarz's inequality and the triangular inequality leads to,

$$\begin{aligned} |x_t^\dagger (\hat{\theta}_{a,t} - \theta_a)| &= \left| x_t^\dagger \left( \lambda_n I + \frac{1}{n} X_{\mathcal{T}}^\dagger X_{\mathcal{T}} \right)^{-1} \left( \frac{1}{n} \sum_{\tau \in \mathcal{T}} x_\tau \epsilon_{a,\tau} - \lambda_n \theta_a \right) \right| \\ &\leq \sqrt{x_t^\dagger \left( \lambda_n I + \frac{1}{n} X_{\mathcal{T}}^\dagger X_{\mathcal{T}} \right)^{-2} x_t} \left( \left| \frac{1}{n} \sum_{\tau \in \mathcal{T}} x_t^\dagger x_\tau \epsilon_{a,\tau} \right| + \lambda_n |x_t^\dagger \theta_a| \right). \end{aligned}$$

We introduce the following notation

$$c_{a,t} \equiv \sqrt{x_t^\dagger \left( \lambda_n I + \frac{1}{n} X_{\mathcal{T}}^\dagger X_{\mathcal{T}} \right)^{-2} x_t}. \quad (8)$$

Note that, given  $a$  and  $t$  both  $n$  and  $\mathcal{T}$  are well specified.

We can now write,

$$\begin{aligned} &\mathbb{P} \left\{ x_t^\dagger \hat{\theta}_{a,t} \geq x_t^\dagger \theta_a + \frac{\Delta_{\min}}{2} \right\} \\ &\leq \mathbb{P} \left\{ \left| \frac{1}{n} \sum_{\tau \in \mathcal{T}} x_t^\dagger x_\tau \epsilon_{a,\tau} \right| \geq \frac{\Delta_{\min}}{2c_{a,t}} - \lambda_n |x_t^\dagger \theta_a| \right\} \\ &\leq \mathbb{P} \left\{ \left| \frac{1}{n} \sum_{\tau \in \mathcal{T}} x_t^\dagger x_\tau \epsilon_{a,\tau} \right| \geq \frac{\Delta_{\min}}{2c_{a,t}} - \lambda_n Q \right\}. \end{aligned}$$

Since  $\epsilon_{a,\tau}$  are sub-gaussian random variables with sub-gaussian constant upper bounded by  $L$  and since  $|x_t^\dagger x_\tau| \leq 1$ , conditioned on  $x_t$ ,  $\mathcal{T}$  and  $\{x_\tau\}_{\tau \in \mathcal{T}}$ , each  $x_t^\dagger x_\tau \epsilon_{a,\tau}$  is a sub-gaussian random variable and together they form a set of i.i.d. sub-gaussian random variables. One can thus apply standard concentration inequality and obtain,

$$\begin{aligned} &\mathbb{P} \left\{ \left| \frac{1}{n} \sum_{\tau \in \mathcal{T}} x_t^\dagger x_\tau \epsilon_{a,\tau} \right| \geq \frac{\Delta_{\min}}{2c_{a,t}} - \lambda_n Q \right\} \\ &\leq \mathbb{E} \left\{ 2e^{-\frac{n}{2L^2} \left( \frac{\Delta_{\min}}{2c_{a,t}} - \lambda_n Q \right)^2} \right\}, \end{aligned} \quad (9)$$

where both  $n$  and  $c_{a,t}$  are random quantities and the operator  $^+$  acts on scalars as follows:  $z^+ = z$  if  $z \geq 0$  and zero otherwise.

We now upper bound  $c_{a,t}$  using the following fact about the eigenvalues of any two real-symmetric matrices  $M_1$  and  $M_2$ :  $\lambda_{\max}((M_1)^{-1}) = 1/\lambda_{\min}(M_1)$  and  $\lambda_{\min}(M_1 + M_2) \geq \lambda_{\min}(M_1) - \lambda_{\max}(M_2) = \lambda_{\min}(M_1) - \|M_2\|$ .

$$\begin{aligned} c_{a,t} &= \sqrt{x_t^\dagger \left( \lambda_n I + \frac{1}{n} X_{\mathcal{T}}^\dagger X_{\mathcal{T}} \right)^{-2} x_t} \\ &\leq \left( \lambda_n + \lambda_{\min}^+(\mathbb{E}\{x_1^\dagger x_1\}) - \left\| \frac{1}{n} X_{\mathcal{T}}^\dagger X_{\mathcal{T}} - \mathbb{E}\{x_1^\dagger x_1\} \right\|^+ \right)^{-1}. \end{aligned}$$

Both the eigenvalue and the norm above only need to be computed over the subspace spanned by the vectors  $x_t$  that occur with non-zero probability. We use the symbol  $^+$  to denote the restriction to this subspace. Now notice that  $\|\cdot\|^+ \leq \|\cdot\|$  and, since we defined  $\Sigma_{\min} \equiv \min_{i: \lambda_i > 0} \lambda_i(\mathbb{E}\{X_1 X_1^\dagger\})$ , we have that  $\lambda_{\min}^+(\mathbb{E}\{X_1 X_1^\dagger\}) \geq \Sigma_{\min}$ . Using the following definition,  $\Delta \Sigma_n \equiv n^{-1} X_{\mathcal{T}}^\dagger X_{\mathcal{T}} - \mathbb{E}\{X_1 X_1^\dagger\}$ , this leads to,  $c_{a,t} \leq (\lambda_n + \Sigma_{\min} - \|\Delta \Sigma_n\|)^{-1} \leq (\Sigma_{\min} - \|\Delta \Sigma_n\|)^{-1}$ .

We now need the following Lemma.

**Lemma 1** *Let  $\{X_i\}_{i=1}^n$  be a sequence of i.i.d. random vectors of 2-norm bounded by 1. Define  $\hat{\Sigma} = \frac{1}{n} \sum_{i=1}^n X_i X_i^\dagger$  and  $\Sigma = \mathbb{E}\{X_1 X_1^\dagger\}$ . If  $\epsilon \in (0, 1)$  then,*

$$\mathbb{P}(\|\hat{\Sigma} - \Sigma\| > \epsilon \|\Sigma\|) \leq 2e^{-C\epsilon^2 n},$$

where  $C < 1$  is an absolute constant.

For a proof see ‘Introduction to the non-asymptotic analysis of random matrices’, 2011, by Roman Vershynin (Corollary 50).

We want to apply this lemma to produce a useful bound on the r.h.s. of (9). First notice that, conditioning on  $n$ , the expression inside the expectation in (9) depends through  $c_{a,t}$  on  $n$  i.i.d. contexts that are distributed according to the original distribution. Because of this, we can write,

$$\begin{aligned} &\mathbb{P} \left\{ \left| \frac{1}{n} \sum_{\tau \in \mathcal{T}} x_t^\dagger x_\tau \epsilon_{a,\tau} \right| \geq \frac{\Delta_{\min}}{2c_{a,t}} - \lambda_n Q \right\} \\ &\leq \mathbb{E} \left\{ 2e^{-\frac{n}{2L^2} \left( \frac{\Delta_{\min}}{2c_{a,t}} - \lambda_n Q \right)^2} \right\} \leq \sum_{n=1}^t \left( \mathbb{P}\{|\mathcal{T}_{a,t-1}| = n\} \right. \\ &\quad \times \mathbb{E} \left\{ 2e^{-\frac{n}{2L^2} \left( \frac{\Delta_{\min}}{2c_{a,t}} - \lambda_n Q \right)^2} \middle| |\mathcal{T}_{a,t-1}| = n \right\} \Big). \end{aligned}$$

Using the following algebraic relation: if  $z, w > 0$  then



$(z - w)^{+2} \geq z^2 - 2zw$ , we can now write,

$$\begin{aligned} & \mathbb{E} \left\{ e^{-\frac{n}{2L^2} \left( \frac{\Delta_{\min}}{2c_{a,t}} - \lambda_n Q \right)^{+2}} \middle| |\mathcal{T}_{a,t-1}| = n \right\} \\ & \leq \mathbb{P} \{ |\Delta \Sigma_n| > \Sigma_{\min}/2 \mid |\mathcal{T}_{a,t-1}| = n \} \\ & + e^{-\frac{n}{2L^2} \left( \frac{\Sigma_{\min} \Delta_{\min}}{4} - \lambda_n Q \right)^{+2}} \\ & \leq \mathbb{P} \{ |\Delta \Sigma_n| > \Sigma_{\min}/2 \mid |\mathcal{T}_{a,t-1}| = n \} \\ & + e^{-\frac{Q \Delta_{\min} \Sigma_{\min}}{4L^2}} e^{-\frac{n(\Delta_{\min})^2(\Sigma_{\min})^2}{32L^2}} \end{aligned}$$

Using Lemma 1 we can continue the chain of inequalities,

$$\begin{aligned} & \mathbb{E} \left\{ e^{-\frac{n}{2L^2} \left( \frac{\Delta_{\min}}{2c_{a,t}} - \lambda_n Q \right)^{+2}} \middle| |\mathcal{T}_{a,t-1}| = n \right\} \\ & \leq 2e^{-C(\Sigma_{\min})^2 n/4} \\ & + e^{-\frac{Q \Delta_{\min} \Sigma_{\min}}{4L^2}} e^{-\frac{n(\Delta_{\min})^2(\Sigma_{\min})^2}{32L^2}}. \end{aligned}$$

Before we proceed we need the following lemma,

**Lemma 2** *If  $n_c = \frac{p}{2k} \log t$  then*

$$\mathbb{P} \{ |\mathcal{T}_{a,t-1}| < n_c \} \leq t^{-\frac{p}{16K}}.$$

We can now write,

$$\begin{aligned} & \mathbb{P} \left\{ \left| \frac{1}{n} \sum_{\tau \in \mathcal{T}} x_t^\top x_\tau \epsilon_{a,\tau} \right| \geq \frac{\Delta_{\min}}{2c_{a,t}} - \lambda_n Q \right\} \\ & \leq \sum_{n=1}^t \mathbb{P} \{ |\mathcal{T}_{a,t-1}| = n \} \\ & \mathbb{E} \left\{ 2e^{-\frac{n}{2L^2} \left( \frac{\Delta_{\min}}{2c_{a,t}} - \lambda_n Q \right)^{+2}} \middle| |\mathcal{T}_{a,t-1}| = n \right\} \\ & \leq \mathbb{P} \{ |\mathcal{T}_{a,t-1}| < n_c \} + 4e^{-C(\Sigma_{\min})^2 n_c/4} \\ & + 2e^{-\frac{Q \Delta_{\min} \Sigma_{\min}}{4L^2}} e^{-\frac{n_c(\Delta_{\min})^2(\Sigma_{\min})^2}{32L^2}} \\ & \leq t^{-\frac{p}{16K}} + 4t^{-\frac{Cp(\Sigma_{\min})^2}{8K}} + 2e^{-\frac{Q \Delta_{\min} \Sigma_{\min}}{4L^2}} t^{-\frac{p(\Delta_{\min})^2(\Sigma_{\min})^2}{64KL^2}}. \end{aligned}$$

We want this quantity to be summable over  $t$ . Hence we require that,

$$p \geq \frac{128KL^2}{(\Delta_{\min})^2(\Sigma_{\min})^2}, p \geq \frac{16K}{C(\Sigma_{\min})^2}, p \geq 32K. \quad (10)$$

It is immediate to see that our proof also follows if  $\Delta_{\min}$ ,  $\Sigma_{\min}$  and  $L$  are replaced by  $\Delta'_{\min} = \min\{1, \Delta_{\min}\}$ ,  $\Sigma'_{\min} = \min\{1, \Sigma_{\min}\}$  and  $L' = \max\{1, L\}$  respectively. If this is done, it is easy to see that conditions (10) are all satisfied by the  $p$  stated in Theorem 2.

Since  $\sum_{t=1}^{\infty} 1/t^2 \leq 2$ , gathering all terms together we have,

$$\begin{aligned} R(T) & \leq p\Delta_{\max}\sqrt{d} + p\Delta_{\max}\sqrt{d} \log T \\ & + \Delta_{\max}\sqrt{d}K \left( 4e^{-\frac{Q \Delta'_{\min} \Sigma'_{\min}}{4L'^2}} + 10 \right) \\ & \leq p\Delta_{\max}\sqrt{d} + 14\Delta_{\max}\sqrt{d}Ke^{Q/4} + p\Delta_{\max}\sqrt{d} \log T. \end{aligned}$$

## 7.2 Proof of Lemma 2

First notice that  $|\mathcal{T}_{a,t-1}| = \sum_{i=1}^{t-1} z_i$  where  $\{z_i\}_{i=1}^{t-1}$  are independent Bernoulli random variables with parameter  $p/(Ki)$ . Remember that we can assume that  $i > p$  since in the beginning of Algorithm 1 we play each arm  $p/K$  times.

Now write,

$$\begin{aligned} \mathbb{P}(|\mathcal{T}_{a,t-1}| < n_c) & = \mathbb{P} \left( \sum_{i=1}^{t-1} z_i < n_c \right) \\ & = \mathbb{P} \left( \sum_{i=1}^{t-1} (z_i - p/(Ki)) < n_c - (p/K) \sum_{i=1}^{t-1} 1/i \right) \\ & \leq \mathbb{P} \left( \sum_{i=1}^{t-1} (-z_i + p/i) > -n_c + (p/K) \sum_{i=1}^{t-1} 1/i \right) \\ & \leq \mathbb{P} \left( \sum_{i=1}^{t-1} (-z_i + p/i) > (p/K) \log t - n_c \right). \quad (11) \end{aligned}$$

Since  $\sum_{i=1}^{t-1} \mathbb{E}\{(z_i - p/(Ki))^2\} = \sum_{i=p+1}^{t-1} (1 - p/(Ki))(p/(Ki)) \leq \frac{p}{K} \log t$ , we have that  $\{-z_i + p/i\}_{i=1}^{t-1}$  are i.i.d. random variables with zero mean and sum of variances upper bounded by  $(p/K) \log t$ . Replacing  $n_c = (p/2K) \log t$  in (11) and applying Bernstein inequality we get,

$$\mathbb{P}(|\mathcal{T}_{a,t-1}| < n_c) \leq e^{-\frac{\frac{1}{2}(p/(2K))^2 \log^2 t}{\frac{p}{K} \log t + \frac{1}{3}(p/(2K)) \log t}} \leq t^{-\frac{p}{16K}}.$$

## 8 Conclusions

We introduced an  $\epsilon$ -greedy type of algorithm that provably achieves logarithmic regret for the contextual multi-armed bandits problem with linear payoffs in the stochastic setting. Our online algorithm is both fast and uses small space. In addition, our bound on the regret scales nicely with dimension of the contextual variables,  $O(\text{poly}(d) \log T)$ . By means of numerical simulations we illustrate how the stochasticity of the contexts is important for our bound to hold. In particular, we show how to construct a scenario for which our algorithm does not give logarithmic regret. The reason for this amounts to the fact that the mean  $n^{-1} X_T^\top X_T$  that is used in estimating the parameters  $\theta_a$  can “forget” previously observed contexts. Because of this, it remains an open problem to show that there are efficient algorithms that achieve  $O(\text{poly}(d) \log T)$  under reward separation ( $\Delta_{\min} > 0$ ) in the non-stochastic setting. We believe that a possible solution might be constructing a variant of our algorithm where in  $n^{-1} X_T^\top X_T$  we use a more careful average of past observed contexts give the current observed context. In addition, we leave it open to produce simple and efficient online algorithms for multi-armed bandit problems under rich context models, like the one we have done here for linear payoff.

## References

- Abbasi-Yadkori, Y., Pál, D., and Szepesvári, C. (2011). Improved algorithms for linear stochastic bandits. *Advances in Neural Information Processing Systems*.
- Auer, P. (2002). Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3.
- Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2):235–256.
- Bartlett, P. and Ben-David, S. (1999). On the hardness of learning with neural networks. In *European Conference on Computational Learning Theory*.
- Chu, W., Li, L., Reyzin, L., and Schapire, R. (2011). Contextual bandits with linear payoff functions. In *Conference on Artificial Intelligence and Statistics (AISTATS)*.
- Dani, V., Hayes, T., and Kakade, S. (2008). Stochastic linear optimization under bandit feedback. In *Proceedings of the 21st Annual Conference on Learning Theory (COLT)*.
- Dudik, M., Hsu, D., Kale, S., Karampatziakis, N., Langford, J., Reyzin, L., and Zhang, T. (2011). Efficient optimal learning for contextual bandits. In *UAI*.
- Johnson, D. and Preparata, F. (1978). The densest hemisphere problem. *Theoretical Computer Science*, 6.
- Langford, J. and Zhang, T. (2007). The epoch-greedy algorithm for contextual multi-armed bandits. *Advances in Neural Information Processing Systems*, 20.
- Li, L., Chu, W., Langford, J., and Schapire, R. (2010). A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670. ACM.
- Rusmevichientong, P. and Tsitsiklis, J. (2010). Linearly parameterized bandits. *Mathematics of Operations Research*, 35(2).
- Sutton and Barto (1998). *Reinforcement learning, and introduction*. Cambridge, MIT Press.